

## 第二次作业

### 第一章

#### 一、名词解释

1. 操作系统：是管理计算机硬件与软件资源的系统软件，作为用户和计算机硬件之间的接口，负责协调和控制计算机系统的运行。
4. 实时操作系统：是一种能够对外部事件或数据做出及时响应，在严格规定的时间内完成对该事件的处理，并控制所有实时任务协调一致地运行。
5. 互斥共享：同一时间段内只允许一个进程或线程访问某类共享资源

#### 三、填空题

1. 硬件系统，软件系统
2. 硬件资源，软件资源
3. 共享性，异步性

### 第二章

#### 一、名词解释

1. 临界区：每个进程中访问临界资源的那段代码
4. 进程同步：异步环境下的一组并发进程因直接制约而互相发送消息，互相合作，互相等待，使得各进程按一定的速度执行的过程

#### 三、填空题

1. 动态性，异步性，结构性
5. 动态，静态
6. 间接相互制约关系
8. 共享存储系统，管道通信系统，消息传递系统
9. 执行，就绪，阻塞

DATE / /

解： semaphore mutex=1, empty=5, apple=0, orange=0;

```

void father() {
    while(1) {
        wait(empty);
        wait(mutex);
        放入一种水果
        signal(mutex);
        if 放入的是苹果 signal(apple);
        else signal(orange);
    }
}

void son() {
    while(1) {
        wait(orange);
        wait(mutex);
        从盘子中取出一个桔子
        signal(mutex);
        signal(empty);
    }
}

void daughter() {
    while(1) {
        wait(apple);
        wait(mutex);
        从盘子中取出一个苹果
        signal(mutex);
        signal(empty);
    }
}

void main() {
    cobegin
        father(); son(); daughter();
    coend
}
    
```

# 五.问答题

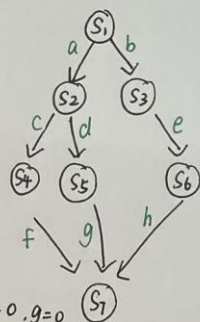
2.解:

$S_1 \rightarrow S_2$ ,  $S_1 \rightarrow S_3$ ,  $S_2 \rightarrow S_4$

$S_2 \rightarrow S_5$ ,  $S_3 \rightarrow S_6$ ,  $S_4 \rightarrow S_7$

$S_5 \rightarrow S_7$ ,  $S_6 \rightarrow S_7$

Semaphore  $a=0, b=0, c=0, d=0, e=0, f=0, g=0, h=0$



```
void S1() {
    signal(a);
    signal(b);
}
```

```
void S2() {
    wait(a);
    signal(c);
    signal(d);
}
```

```
void S3() {
    wait(b);
    signal(e);
}
```

```
void S4() {
    wait(c);
    signal(f);
}
```

```
void S5() {
    wait(d);
    signal(g);
}
```

```
void S6() {
    wait(e);
    signal(h);
}
```

```
void S7() {
    wait(f);
    wait(g);
    wait(h);
}
```

```
void main() {
    wbegin
    S1(); S2(); S3(); S4(); S5();
    S6(); S7();
    wend
}
```

# 五.问答题

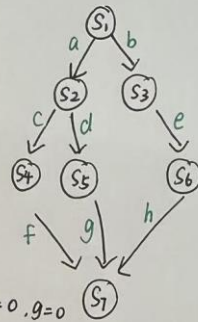
2.解:

$S_1 \rightarrow S_2, S_1 \rightarrow S_3, S_2 \rightarrow S_4$

$S_2 \rightarrow S_5, S_3 \rightarrow S_6, S_4 \rightarrow S_7$

$S_5 \rightarrow S_7, S_6 \rightarrow S_7$

Semaphore  $a=0, b=0, c=0, d=0, e=0, f=0, g=0, h=0$



```
void S1() {
    signal(a);
    signal(b);
}
```

```
void S2() {
    wait(a);
    signal(c);
    signal(d);
}
```

```
void S3() {
    wait(b);
    signal(e);
}
```

```
void S4() {
    wait(c);
    signal(f);
}
```

```
void S5() {
    wait(d);
    signal(g);
}
```

```
void S6() {
    wait(e);
    signal(h);
}
```

```
void S7() {
    wait(f);
    wait(g);
    wait(h);
}
```

```
void main() {
    wbegin
    S1(); S2(); S3(); S4(); S5();
    S6(); S7();
    wend
}
```

5. 解: semaphore seats = N, mutex = 1, service = 0

void customer () {

P(seats);

找座坐下;

V(service);

等待;

P(mutex);

剪头;

V(mutex);

付费;

V(seats);

}

void barber () {

while (1) {

P(service);

剪头;

}

}

Request  
不能给资源分配

No.

DATE / /

### 第三章

#### 一、名词解释

2. 处理机调度：对处理机进行分配，调度的实质是资源分配

3. 周转时间：周转时间 = 完成时间 - 到达时间

4. 死锁：P<sub>1</sub>、P<sub>2</sub> 双方都希望对方释放自己所需的资源，一直僵持下去。

#### 二、填空题

2. 提交，后备，完成 6. 预防死锁，避免死锁，检测死锁，解除死锁

#### 五、问答题

1. 解：FCFS

进程	完成时间	周转时间	带权周转时间	平均周转时间 = (3+7+9+12+12)/5 = 8.6
A	3	3	1	
B	9	7	7/6	
C	13	9	9/4	
D	18	12	12/5	
E	20	12	12/6	

平均带权周转时间 = (1+1.17+2.25+2.4+6)/5 = 2.564

非抢占式 SJF

进程	完成时间	周转时间	带权周转时间	平均周转时间 = (3+7+11+14+3)/5 = 7.6
A	3	3	1	
B	9	7	7/6	
C	15	11	11/4	
D	20	14	14/5	
E	11	3	3/2	

平均带权周转时间 = (1+1.17+2.75+2.8+1.5)/5 = 1.844

抢占式 SJF

进程	完成时间	周转时间	带权周转时间	平均周转时间 = (3+13+4+14+2)/5 = 7.2
A	3	3	1	
B	15	13	13/6	
C	8	4	4	
D	20	14	14/5	
E	10	2	1	

平均带权周转时间 = (1+2.17+1+2.8+1)/5 = 1.544

HRRN

0时刻，执行A

3时刻，B:  $\frac{1+6}{6} = 1.17$  执行B

9时刻，C:  $\frac{5+4}{4} = 2.25$ ，D:  $\frac{3+5}{5} = 1.6$ ，E:  $\frac{1+2}{2} = 1.5$  执行C

13时刻，D:  $\frac{7+5}{5} = 2.4$ ，E:  $\frac{5+2}{2} = 3.5$  执行D

15时刻，执行D

进程	完成时间	周转时间	带权周转时间	平均周转时间 = (3+7+9+14+7)/5 = 8
A	3	3	1	
B	9	7	7/6	
C	13	9	9/4	
D	20	14	14/5	
E	15	7	7/2	

平均带权周转时间 = (1+1.17+2.25+2.8+3.5)/5 = 2.144

No.

DATE

RR 时间片为 1

[A] [A]

0 1 2

就绪队列: [B] [A]

[A] [A] [A]

0 1 2 3 4

[B] [C]

[A] [A] [B] [A] [B] [C]

0 1 2 3 4 5 6

[B] [D] [C]

[A] [A] [B] [A] [B] [C] [B] [D]

0 1 2 3 4 5 6 7 8

[C] [B] [E] [D]

[A] [A] [B] [A] [B] [C] [B] [D] [C] [B] [E] [D] [C] [B] [D] [D]

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

A完成

B完成 C完成 D完成

进程	完成时间	周转时间	带权周转时间	平均周转时间 = 10.8
A	4	4	4/3	
B	18	16	16/6	平均带权周转时间 = 2.71
C	17	13	13/4	
D	20	14	14/5	
E	15	7	7/2	



2.

解:

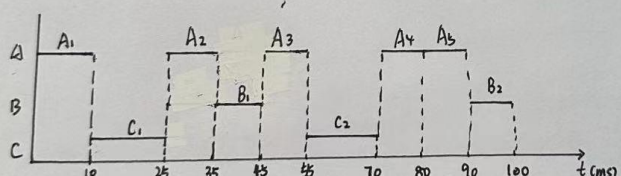
0时刻:  $A_1$  的松弛度为  $20ms - 10ms - 0ms = 10ms$   
 $B_1$  的松弛度为  $50ms - 10ms - 0ms = 40ms$   
 $C_1$  的松弛度为  $50ms - 15ms - 0ms = 35ms$  }  $A_1$  最紧急, 执行  $A_1$

10ms时刻: 任务  $A$  尚未进入第2周期  
 $B_1: 50ms - 10ms - 10ms = 30ms$   
 $C_1: 50ms - 15ms - 10ms = 25ms$  } 执行  $C_1$

25ms时刻:  $A_2: 40ms - 10ms - 25ms = 5ms$   
 $B_1: 50ms - 10ms - 25ms = 15ms$   
 任务  $C$  尚未进入第2周期 } 执行  $A_2$

35ms时刻: 任务  $A$  尚未进入第3周期  
 $B_1: 50ms - 10ms - 35ms = 5ms$   
 任务  $C$  尚未进入第2周期 } 执行  $B_1$

45ms时刻:  $A_3: 60ms - 10ms - 45ms = 5ms$   
 任务  $B$  和任务  $C$  尚未进入第2周期 } 执行  $A_3$



55ms:  $A$  尚未进入第4周期  
 $B_2: 100ms - 10ms - 55ms = 35ms$   
 $C_2: 100ms - 15ms - 55ms = 30ms$  } 执行  $C_2$

70ms:  $A_4: 80ms - 10ms - 70ms = 0ms$   
 $B_2: 100ms - 10ms - 70ms = 20ms$   
 任务  $C$  尚未进入第3周期 } 执行  $A_4$

80ms:  $A_5: 100ms - 10ms - 80ms = 10ms$   
 $B_2: 100ms - 10ms - 80ms = 10ms$   
 任务  $C$  尚未进入第3周期 } 当剩余松弛时间一样时, 选短作业优先, 执行  $A_5$

90ms: 任务  $A$  尚未进入第6周期  
 $B_2: 100ms - 10ms - 90ms = 0ms$   
 任务  $C$  尚未进入第3周期 } 执行  $B_2$

3. 解: (1) 先给  $P_0$   $Available = (1, 6, 2, 2) - (0, 0, 1, 2) + (0, 0, 4, 4) = (1, 6, 5, 4)$

给  $P_3$   $Available = (1, 6, 5, 4) - (0, 6, 5, 2) + (0, 9, 8, 4) = (1, 9, 8, 6)$

给  $P_1$   $Available = (1, 9, 8, 6) - (1, 7, 5, 0) + (2, 7, 5, 0) = (2, 9, 8, 6)$

给  $P_2$   $Available = (2, 9, 8, 6) - (2, 3, 5, 6) + (3, 6, 10, 10) = (3, 12, 13, 10)$

给  $P_4$   $Available = (3, 12, 13, 10) - (0, 6, 5, 6) + (0, 6, 6, 10) = (3, 12, 14, 14)$

安全, 可以找到安全序列  $\{P_0, P_3, P_1, P_2, P_4\}$

(2)  $Request(1, 2, 2, 2) \leq Need_2(2, 3, 5, 6)$

$Request(1, 2, 2, 2) \leq Available(1, 6, 2, 2)$

系统暂时假定可为  $P_2$  分配资源, 修改  $Available$  变成  $(0, 4, 0, 0)$

可用资源  $Available(0, 4, 0, 0)$  已不能满足任何进程的需要, 故系统进入不安全状态

$\therefore$  系统不能将资源分配给它

(3) 会进入死锁状态